

**SISTEM PENDETEKSI JATUH PADA MANUSIA BERBASIS
PENGOLAHAN CITRA**

SKRIPSI

untuk memenuhi salah satu persyaratan
mencapai derajat Sarjana S1



Disusun oleh:

Mohd Brado Frasetyo

14524030

**Jurusan Teknik Elektro
Fakultas Teknologi Industri
Universitas Islam Indonesia
Yogyakarta
2018**

LEMBAR PENGESAHAN PEMBIMBING

SISTEM PENDETEKSI JATUH PADA MANUSIA BERBASIS PENGOLAHAN CITRA



Yogyakarta, 05 Februari 2018

Menyetujui,

Pembimbing

Elvira Sukma Wahyuni, S.Kom., M.Eng

155231301

PERNYATAAN

Saya yang bertanda tangan dibawah ini :

Nama : Mohd Brado Frasetyo

NIM : 14524030

Menyatakan bahwa :

1. Skripsi ini tidak mengandung karya yang diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak mengandung karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.
2. Informasi dan materi Skripsi yang terkait hak milik, hak intelektual, dan paten merupakan milik bersama antara tiga pihak yaitu penulis, dosen pembimbing, dan Universitas Islam Indonesia. Dalam hal penggunaan informasi dan materi Skripsi terkait paten maka akan diskusikan lebih lanjut untuk mendapatkan persetujuan dari ketiga pihak tersebut diatas.

Yogyakarta, 05 Februari 2018



Mohd Brado Frasetyo

LEMBAR PENGESAHAN PENGUJI

SISTEM PENDETEKSI JATUH PADA MANUSIA BERBASIS PENGOLAHAN CITRA



Tim Penguji :

Ketua

Elvira Sukima Wahyuni, S.Pd., M.Eng.

Anggota 1

Alvin Sahroni, S.T., M.Eng., PhD.

Anggota 2

Dwi Ana Ratnawati, S.T., M.Eng.

Mengetahui,



Dr.Eng. Hendra Setiawan, S.T., M.T.

KATA PENGANTAR



Assalamu'alaikum Wr.Wb

Puji syukur penyusun panjatkan atas kehadiran Allah SWT, yang telah memberikan rahmat dan karunianya sehingga penulis dapat menyelesaikan laporan tugas akhir ini sebagai salah satu syarat memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia.

Sholawat beserta salam senantiasa penyusun haturkan kepada junjungan kita Nabi besar Muhammad SAW yang telah membimbing umat dari zaman kegelapan menuju zaman yang sesuai dengan ketentuan Allah SWT. Selama pelaksanaan tugas akhir penyusun banyak mendapat bantuan dari berbagai pihak, untuk itu dengan kerendahan hati, penyusun ingin berterima kasih kepada :

1. Orang tua tercinta Ibu Leni Edmizar, SE. dan Bapak Hamka Makmur, SH. yang selalu mendo'akan dan membantu segala sesuatunya.
2. Ibu Elvira Sukma Wahyuni, S.Pd.,M.Eng. selaku Dosen Pembimbing tugas akhir yang telah mendampingi dan memberikan bantuan pikiran dan materi.
3. Bapak Dr. Eng. Hendra Setiawan, S.T.,M.Eng. selaku Ketua Jurusan Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia.
4. Saudara M.Zharfan Wiranata, Mayong N. Saverio, dan teman-teman yang lainnya yang telah berbaik hati membantu pelaksanaan tugas akhir ini sehingga dapat diselesaikan.
5. Saudari Yosi May Tesa, yang selalu menjadi rekan terbaik selama masa kuliah.

Saya menyadari sepenuhnya bahwa dalam pelaksanaan tugas akhir ini banyak kekeliruan dan kekurangan, untuk itu saya mohon maaf sebesar-besarnya semoga hasil dari tugas akhir ini dapat bermanfaat bagi semua pihak yang membutuhkan di masa yang akan datang, Amin.

Yogyakarta, 05 Februari 2018

(Mohd Brado Frasetyo)

14524030

ARTI LAMBANG DAN SINGKATAN

MHI	: <i>Motion History Image</i>
GUI	: <i>Graphical User Interface</i> .
$f(x,y)$: Koordinat citra digital.
x	: Posisi piksel pada koordinat x.
y	: Posisi piksel pada koordinat y.
H_τ	: <i>Motion History Image</i> (MHI).
$D(x,y,t)$: <i>Binary sequense</i> gerakan objek.
m_{pq}	: Nilai <i>Moment</i> .
x	: Posisi piksel pada koordinat x.
y	: Posisi piksel pada koordinat y.
\bar{x}	: <i>Centroid</i> pada koordinat sumbu x.
\bar{y}	: <i>Centroid</i> pada koordinat sumbu y.
μ_{pq}	: <i>Central moments</i> .
θ	: Orientasi <i>ellipse</i> .
J	: <i>covariance matrix</i> .
I_{min}	: Momen inersia terkecil.
I_{max}	: Momen inersia terbesar.
a	: <i>Major semi-axis</i> a.
b	: <i>Minor semi-axis</i> b.
ρ	: Rasio <i>ellipse</i> .
C_{motion}	: Presentase pergerakan objek.
$Blob$: Piksel yang merepresentasikan objek.
Σ_θ	: Standar deviasi dari nilai <i>theta</i> .
Σ_ρ	: Standar deviasi dari nilai <i>rho</i> .

ABSTRAK

Peningkatan jumlah lansia di Indonesia cukup tinggi, suatu hal yang harus diperhatikan adalah terdapat lansia yang tinggal sendiri tanpa ada keluarga di rumah. Hal ini memiliki resiko tinggi terutama untuk aspek biologis, yaitu apabila terjadi sesuatu yang tidak diinginkan salah satunya adalah jatuh, sehingga dibutuhkan sistem pendeteksi jatuh untuk *monitoring* keadaan lansia di rumah. Pada penelitian ini telah dirancang sistem pendeteksi jatuh pada manusia menggunakan pengolahan citra dengan input dari kamera CCTV, metode *Motion History Image* (MHI) dan *Approximated Ellipse*. Metode tersebut akan menghasilkan nilai parameter C_motion , $Sigma_theta$, dan $Sigma_rho$ yang akan digunakan sebagai acuan pendeteksi jatuh. Hasil dari penelitian ini menunjukkan akurasi sebesar 83.33% untuk data kondisi jatuh maupun tidak jatuh.

Kata Kunci : Pengolahan citra, Kamera CCTV, *Motion History Image*, *Approximated Ellipse*.

DAFTAR ISI

LEMBAR PENGESAHAN PEMBIMBING	ii
PERNYATAAN.....	iii
LEMBAR PENGESAHAN PENGUJI	iv
KATA PENGANTAR.....	v
ARTI LAMBANG DAN SINGKATAN	vi
ABSTRAK	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR	x
DAFTAR TABEL.....	xi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	2
BAB 2 TINJAUAN PUSTAKA	3
2.1 Studi Literatur	3
2.2 Tinjauan Teori.....	3
2.2.1 Representasi Citra Digital	3
2.2.2 Jenis Citra Berdasarkan Warna.....	4
2.2.3 <i>Motion History Image</i> (MHI)	5
2.2.4 <i>Approximated Ellipse</i>	5
BAB 3 METODOLOGI.....	8
3.1 Alur Penelitian	8

3.1.1 Pengambilan Data Video	8
3.1.2 Algoritma Sistem Pendeteksi Jatuh	8
3.1.3 Menampilkan Hasil & Analisa	9
3.2 Sistem Pendeteksi Jatuh.....	10
3.2.1 Input Data Video.....	11
3.2.2 <i>Pre-Processing</i>	11
3.2.3 Membangun <i>Motion History Image</i> (MHI)	12
3.2.4 Mencari nilai ρ dan ϕ	12
3.2.5 Mencari nilai C_{motion} , Σ_{ρ} , dan Σ_{θ}	12
BAB 4 HASIL DAN PEMBAHASAN.....	14
4.1 Pengujian Sistem Pendeteksi Jatuh.....	14
4.1.1 Pengujian Kamera CCTV	14
4.1.2 Pengujian Perhitungan Parameter Pergerakan Objek	14
4.1.3 Pengujian Algoritma Peneteksi Jatuh	15
4.2 Pengujian Algoritma Terhadap Kondisi Berbeda.	17
4.2.1 Pengujian Algoritma Pendeteksi Jatuh Terhadap Posisi Jatuh yang Berbeda.	17
4.2.2 Pengujian Algoritma Pendeteksi Jatuh Terhadap Gerakan Jatuh yang Berbeda..	19
4.2.3 Pengujian Algoritma Pendeteksi Jatuh Terhadap Intensitas Cahaya Berbeda.	21
4.2.4 Pengujian Algoritma Pendeteksi Jatuh Terhadap Warna Objek Berbeda	23
4.2.5 Pengujian Algoritma Pendeteksi Jatuh Terhadap Objek Berbeda.....	24
4.2.6 Pengujian Algoritma Pendeteksi Jatuh Terhadap Kecepatan Jatuh Berbeda.	25
4.3 Pengujian Akurasi Sistem Pendeteksi Jatuh.	26
BAB 5 KESIMPULAN DAN SARAN.....	28
5.1 Kesimpulan	28
5.2 Saran	28
DAFTAR PUSTAKA	29
LAMPIRAN	30

DAFTAR GAMBAR

Gambar 2.1 Koordinat Citra Digital.....	4
Gambar 2.2 <i>Motion History Image</i>	5
Gambar 2.3 Ilustrasi nilai θ dan ρ pada sebuah <i>ellipse</i>	7
Gambar 3.1 Diagram Blok Alur Penelitian.....	8
Gambar 3.2 Diagram Alir Algoritma Pendeteksi Jatuh	9
Gambar 3.3 Diagram Alir Sistem Deteksi Jatuh.....	10
Gambar 3.4 Hasil <i>Foreground Detection</i> pada Sebuah Citra	11
Gambar 3.5 Hasil <i>Filtering Blob</i> pada Citra <i>Foreground</i>	11
Gambar 3.6 Hasil citra <i>Motion History Image</i> (MHI).	12
Gambar 4.1 Tampilan Objek dari Kamera CCTV	14
Gambar 4.2 Hasil Parameter Sistem Pendeteksi Jatuh.....	15
Gambar 4.3 Tampilan Hasil Parameter Sistem Pendeteksi Jatuh	16
Gambar 4.4 Perubahan parameter ketika objek berada di posisi tengah ruangan.....	17
Gambar 4.5 Perubahan parameter ketika objek dekat dengan kamera	18
Gambar 4.6 Perubahan parameter ketika jarak terlalu jauh	19
Gambar 4.7 Perubahan parameter ketika objek terhalang benda lain	19
Gambar 4.8 Perubahan parameter ketika jatuh arah depan, samping, dan belakang	20
Gambar 4.9 Perubahan parameter ketika intensitas cahaya terang	22
Gambar 4.10 Perubahan parameter ketika intensitas cahaya redup	22
Gambar 4.11 Perubahan parameter ketika intensitas cahaya gelap	23
Gambar 4.12 Perubahan parameter ketika warna objek berbeda	24
Gambar 4.13 Perubahan parameter ketika objek berbeda.....	25
Gambar 4.14 Perubahan parameter ketika kecepatan jatuh berbeda.....	26

DAFTAR TABEL

Tabel 4.1 Hasil pengujian pendeteksi jatuh terhadap posisi jatuh objek yang berbeda	17
Tabel 4.2 Hasil pengujian pendeteksi jatuh terhadap pergerakan jatuh yang berbeda	20
Tabel 4.3 Hasil pengujian pendeteksi jatuh terhadap intensitas cahaya yang berbeda	21
Tabel 4.4 Hasil pengujian pendeteksi jatuh terhadap warna objek yang berbeda.....	23
Tabel 4.5 Hasil pengujian pendeteksi jatuh terhadap objek yang berbeda	24
Tabel 4.6 Hasil pengujian pendeteksi jatuh terhadap kecepatan jatuh yang berbeda	25
Tabel 4.7 Hasil pengujian akurasi sistem pendeteksi jatuh.....	26

BAB 1

PENDAHULUAN

1.1 Latar Belakang Masalah

Indonesia merupakan salah satu negara yang mengalami peningkatan jumlah lansia (Lanjut Usia) yang cukup tinggi, dimana pada tahun 2100 populasi lansia di Indonesia diprediksi lebih tinggi dari rata-rata populasi lansia dunia[1]. Sesuatu yang harus diperhatikan adalah terdapat 9.66% lansia tinggal sendiri tanpa ada keluarga di rumah, hal ini memiliki resiko tinggi terutama untuk aspek biologis, karena aspek biologis berpengaruh pada penurunan daya tahan fisik dan rentan terhadap penyakit[2]. Sehingga tidak diketahui apabila terjadi sesuatu yang tidak diinginkan menimpa lansia, dari kondisi tersebut maka dibutuhkan sistem yang dapat mendeteksi pergerakan seseorang terutama untuk pergerakan yang tidak biasa seperti terjatuh.

Terdapat beberapa metode yang pernah digunakan dalam membuat sistem pendeteksi jatuh, seperti pemasangan piranti berupa sensor pada tubuh objek [3], ataupun dengan menggunakan kamera [4]. Untuk kondisi di Indonesia, pendeteksi jatuh dengan menggunakan sensor kurang efektif, faktor ingatan lansia yang biasanya lemah sehingga sensor memiliki resiko rusak atau hilang yang cukup tinggi. Selain itu, rata-rata penduduk indonesia merupakan muslim [5], sehingga sensor akan sering dilepas ketika ingin melakukan ibadah. Pendeteksi jatuh menggunakan kamera merupakan salah satu solusi untuk mendeteksi objek, karena dapat menganalisa kondisi objek tanpa bersentuhan secara langsung. Metode ini menggunakan pemrosesan citra yang memproses citra atau gambar yang diambil dari kamera kemudian mendeteksi pergerakan berdasarkan perubahan piksel dari citra tersebut.

Motion History Image (MHI) merupakan salah satu metode pemrosesan citra untuk mendeteksi pergerakan seseorang, video yang terekam dibagi menjadi beberapa gambar sehingga membentuk gambar sekuens, dari banyak gambar ini akan diproses untuk mengetahui pergerakan objek yaitu dari perubahan pikselnya, ketika pergerakan kegiatan normal piksel citra tidak mengalami perubahan yang besar, sedangkan ketika pergerakan yang tidak normal seperti jatuh, pergerakan objek berubah dari kondisi berdiri kemudian duduk atau tidur kemudian diam, piksel akan mengalami perubahan yang sangat cepat [4].

1.2 Rumusan Masalah

Berdasarkan latar belakang sebelumnya, maka rumusan masalah dari penelitian ini adalah Bagaimana merancang sistem pendeteksi jatuh pada manusia menggunakan pemrosesan citra dengan metode *Motion History Image* (MHI) dan kamera CCTV?

1.3 Batasan Masalah

Dalam penelitian ini terdapat beberapa batasan masalah dalam menyelesaikan penelitian ini. Diantaranya adalah :

1. Kamera CCTV yang digunakan berjumlah 4 buah, kemudian dipasang pada setiap sudut ruangan
2. Kondisi intensitas cahaya ruangan harus memadai, sehingga kamera CCTV dapat mengambil gambar objek yang akan dideteksi dengan jelas.
3. Objek yang berada didalam ruangan berjumlah satu orang.
4. Algoritma sistem digunakan untuk data offline (rekaman video).

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah mendapatkan sistem yang dapat mendeteksi jatuh pada manusia menggunakan pemrosesan citra dengan metode *Motion History Image* (MHI) dan kamera CCTV.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah sistem yang dirancang dapat membantu dalam mencegah kondisi yang tidak diinginkan pada lansia ketika mengalami jatuh, terutama bagi lansia yang berada di rumah seorang diri tanpa adanya pengawasan dari keluarga.

BAB 2

TINJAUAN PUSTAKA

2.1 Studi Literatur

Sistem deteksi jatuh menggunakan *Motion History image* (MHI) ini sudah pernah dilakukan oleh beberapa peneliti sebelumnya, adapun beberapa penelitian tersebut sebagai berikut.

Penelitian dilakukan oleh Caroline Rougier dkk, penelitian ini tentang bagaimana mendeteksi jatuh seseorang dengan menggunakan *computer vision system*. Metode yang digunakan adalah dengan menggunakan *Motion History Image* (MHI) dan *Human Shape Variation* yang mendeteksi perubahan piksel gambar *video sequence* sehingga dapat mendeteksi perubahan gerakan seseorang, ketika pergerakan sangat cepat dan tiba-tiba berhenti maka seseorang dianggap jatuh. Hasilnya adalah sistem dapat mendeteksi orang terjatuh dan dapat membedakan dengan pergerakan sehari-hari. Akan tetapi sistem ini hanya menggunakan 1 kamera sehingga tidak semua gerakan objek dapat dideteksi dengan baik [4].

Penelitian lain dilakukan oleh Nuttapon Worrakulpanit dkk, penelitian ini tentang bagaimana mendeteksi jatuh seseorang dengan melihat perubahan kecepatan gerakan dan perubahan posisi tubuh objek. Metode yang digunakan adalah *Motion History Image* (MHI), kemudian membuat algoritma deteksi jatuh dengan perubahan dua parameter yaitu C_{motion} dan $Sigma_{theta}$. Akan tetapi nilai $Sigma_{theta}$ tidak cukup untuk mendeteksi perubahan posisi tubuh seseorang karena hanya mendeteksi perubahan dengan gerakan sejajar dengan sumbu optik kamera [6].

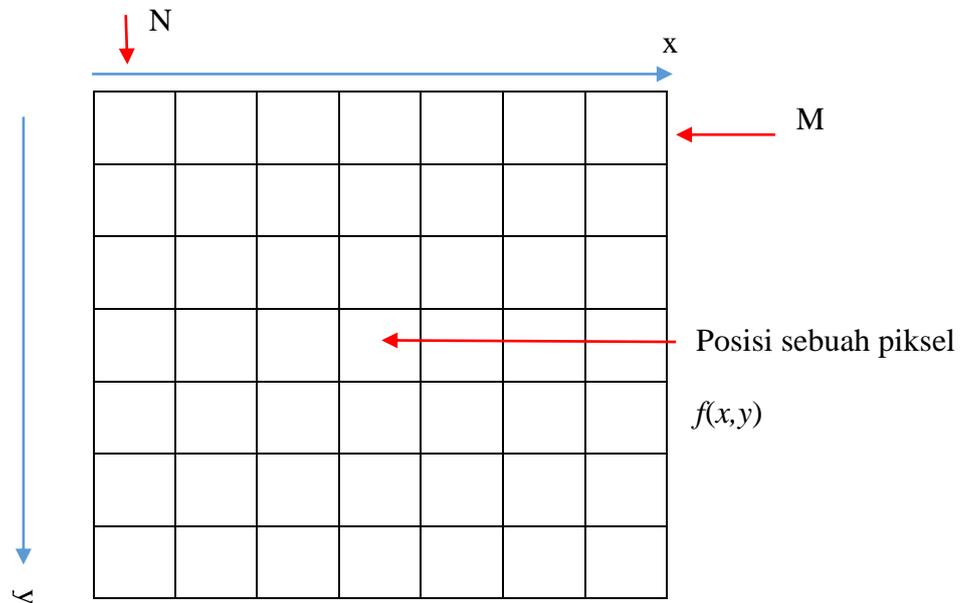
Penelitian lainnya dilakukan oleh Suad Albawendi dkk, penelitian ini juga menggunakan *Motion History Image* (MHI) sebagai pendeteksi gerakan objek. Akan tetapi pada penelitian ini mereka menggunakan kamera yang diletakkan persis didepan objek sehingga jarak pandang kamera yang mereka gunakan terbatas [7].

2.2 Tinjauan Teori

2.2.1 Representasi Citra Digital

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra sebagai keluaran suatu sistem perekaman data dapat bersifat digital ataupun bersifat analog. Citra digital adalah citra yang dapat diolah oleh komputer, sedangkan citra analog adalah citra yang bersifat kontinyu yang apabila ingin diolah harus dikonversikan terlebih dahulu menjadi citra

digital. Citra digital didapatkan melalui proses akuisisi dengan menggunakan sensor optik, setelah proses akuisisi citra digital membentuk bidang dua dimensi (x,y) . Besar intensitas yang diterima sensor di setiap titik (x,y) disimbolkan oleh $f(x,y)$ dan besarnya tergantung pada intensitas yang dipantulkan oleh objek [8]. Citra digital berukuran M baris dan N kolom, Gambar 2.1 menunjukkan koordinat citra digital.



Gambar 2.1 Koordinat citra digital.

2.2.2 Jenis Citra Berdasarkan Warna

Terdapat beberapa jenis citra digital berdasarkan warna, beberapa jenis citra digital yang sering digunakan adalah citra *biner*, citra *grayscale*, dan citra warna.

1. Citra *Biner*

Citra biner hanya memiliki 2 warna yaitu hitam dan putih (0 dan 1), dibutuhkan 1 bit di memori untuk menyimpan kedua warna untuk citra jenis ini [8].

2. Citra *Grayscale*

Citra *grayscale* memiliki jumlah warna tergantung dari jumlah bit yang disediakan di memori untuk menampung kebutuhan warna ini. Jumlah warna didapatkan dari 2^n dimana n merupakan jumlah bit memori, misalkan citra 2 bit berarti mewakili 4 gradasi warna, dan citra 8 bit mewakili 256 gradasi warna keabuan [8].

3. Citra warna

Setiap piksel pada citra warna mewakili warna yang merupakan kombinasi dari tiga warna dasar (RGB = *Red Green Blue*). Setiap warna dasar menggunakan penyimpanan 8 bit = 1 *byte*, yang berarti setiap warna mempunyai gradasi sebanyak 256 warna. Berarti setiap piksel mempunyai kombinasi warna sebanyak $2^8 \times 2^8 \times 2^8 = 16.581.375$ warna [8].

2.2.3 Motion History Image (MHI)

Motion History Image adalah metode pengolahan citra untuk mendeteksi pergerakan objek, metode ini mendeteksi pergerakan berdasarkan perubahan piksel. Pergerakan objek diambil dari video yang sudah dibagi menjadi beberapa gambar sekuens. *Motion History Image* (MHI) ini pertama kali diperkenalkan oleh Bobick dan Davis [9] . Untuk mendapatkan MHI, langkah pertama adalah mengekstrak *binary sequense* gerakan objek $D(x,y,t)$ dari gambar asli $I(x,y,t)$ menggunakan metode *image-differencing* [4]. Kemudian setiap piksel dari *Motion History Image* H_t merupakan fungsi dalam rentang waktu t ($1 \leq \tau \leq n$), persamaannya adalah :

$$H_{\tau}(x,y,t) = \begin{cases} \tau & \text{if } D(x,y,t) = 1 \\ \max(0, H_{\tau}(x,y,t-1) - 1) & \text{otherwise} \end{cases} \quad (2.1)$$

Keterangan :

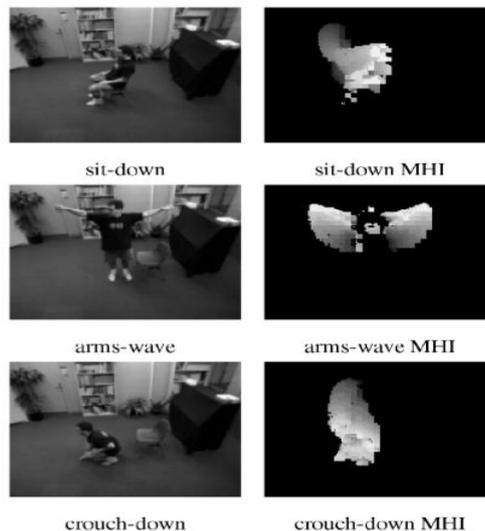
x = Posisi piksel pada koordinat x.

y = Posisi piksel pada koordinat y.

$H_{\tau}(x,y,t)$ = *Motion History Image* (MHI).

$D(x,y,t)$ = *Binary sequense* gerakan objek.

Hasilnya adalah nilai skalar dari gambar dimana pergerakan dari objek akan berubah menjadi warna putih, berikut merupakan hasil dari *Motion History Image* (MHI) :



Gambar 2.2 *Motion History Image* [9].

2.2.4 Approximated Ellipse

Objek akan diprediksi menggunakan *ellipse* menggunakan *moments*. Sebuah *ellipse* didefinisikan oleh *centroid* (x,y) , orientasi, dan *major semi-axis* a dan *minor semi-axis* b [4].

Untuk sebuah citra kontinyu, nilai *moments* didapatkan dari :

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \quad (2.2)$$

Keterangan :

m_{pq} = Nilai *Moments*.

Nilai p,q = 0,1,2

Centroid dari *ellipse* didapatkan dengan cara menghitung koordinat dari pusat massa dengan *first and zero spatial order moments* :

$$\bar{x} = \frac{m_{10}}{m_{00}}; \bar{y} = \frac{m_{01}}{m_{00}} \quad (2.3)$$

Keterangan :

\bar{x} = *Centroid* pada koordinat sumbu x.

\bar{y} = *Centroid* pada koordinat sumbu y.

Centroid (x,y) digunakan untuk menghitung *central moments* :

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \bar{x})^p (y - \bar{y})^q f(x, y) dx dy \quad (2.4)$$

Keterangan :

μ_{pq} = *Central moments*.

Sudut antara sumbu utama objek dan sumbu horizontal x memberikan orientasi dari *ellipse*, dan bisa dihitung dengan *central moments* orde 2:

$$\theta = \frac{1}{2} \tan^{-1} \left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right) \quad (2.5)$$

Keterangan :

θ = Orientasi *ellipse*.

Untuk mendapatkan *major semi-axis* a dan *minor semi-axis* b dari *ellipse*, kita harus menghitung *Imin* dan *Imax*, yaitu momen inersia terkecil dan momen inersia terbesar [4]. Nilai ini dapat dihitung dengan mengevaluasi *eigenvalue* dari *covariance matrix*:

$$J = \begin{pmatrix} \mu_{20} & \mu_{11} \\ \mu_{11} & \mu_{02} \end{pmatrix} \quad (2.6)$$

Eigenvalue dari matrix J adalah I_{min} dan I_{max} yang dihitung dengan :

$$I_{min} = \mu_{20} + \mu_{02} - \frac{\sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}}{2} \quad (2.7)$$

$$I_{max} = \mu_{20} + \mu_{02} + \frac{\sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}}{2}$$

Keterangan :

J = covariance matrix.

I_{min} = Momen inersia terkecil.

I_{max} = Momen inersia terbesar.

Kemudian *major semi-axis* a dan *minor semi-axis* b dari *ellipse* adalah :

$$a = \left(\frac{4}{\pi}\right)^{\frac{1}{4}} \left(\frac{(I_{max})^3}{I_{min}}\right)^{\frac{1}{8}} \quad (2.8)$$

$$b = \left(\frac{4}{\pi}\right)^{\frac{1}{4}} \left(\frac{(I_{min})^3}{I_{max}}\right)^{\frac{1}{8}}$$

Dengan a dan b kita dapat menentukan rasio dari *ellipse*:

$$\rho = \frac{a}{b} \quad (2.9)$$

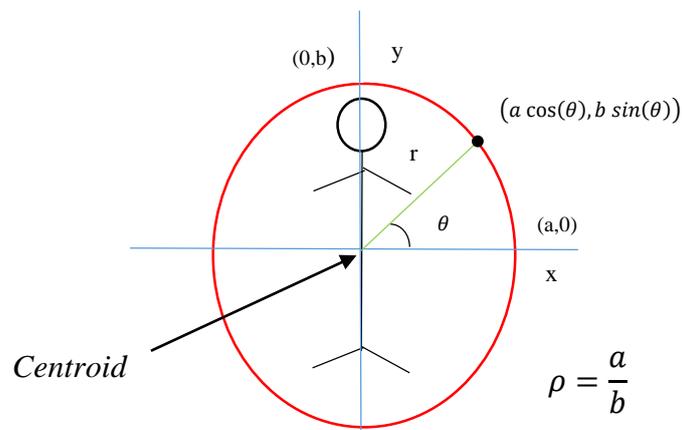
Keterangan :

a = Major semi-axis a .

b = Minor semi-axis b .

ρ = Rasio *ellipse*.

Gambar 2.3 berikut merupakan ilustrasi nilai θ dan ρ pada sebuah *ellipse*.



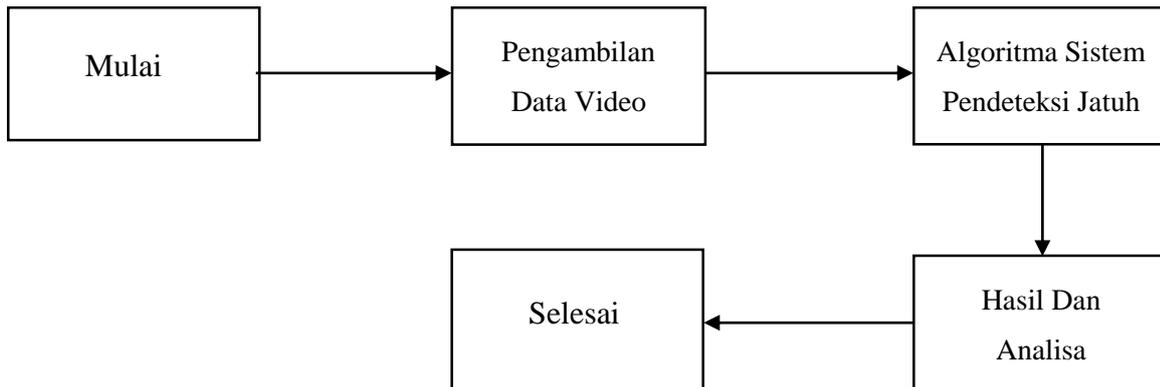
Gambar 2.3 Ilustrasi *approximated ellipse* pada objek.

BAB 3

METODOLOGI

3.1 Alur Penelitian

Penelitian sistem pendeteksi jatuh pada manusia ini memiliki beberapa tahap, dimana setiap tahap memiliki fungsi masing-masing untuk menyelesaikan sistem. Gambar 3.1 merupakan diagram blok penelitian tugas akhir ini.



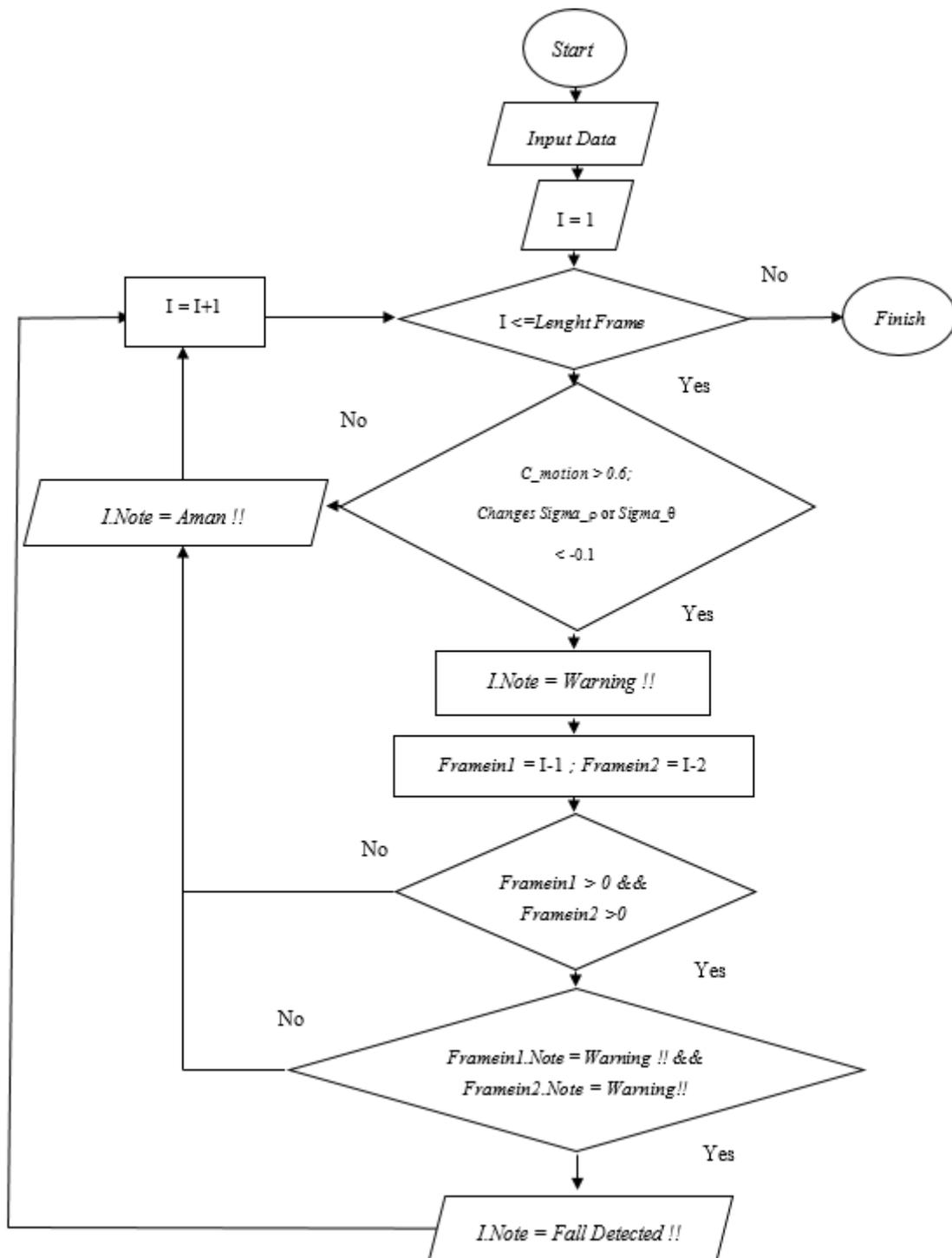
Gambar 3.1 Diagram Blok Alur Penelitian.

3.1.1 Pengambilan Data Video

Tahap pertama dalam penelitian adalah mengambil data video sebagai data input pendeteksi jatuh menggunakan kamera CCTV yang telah dipasang pada setiap sudut ruangan admin Laboratorium Pemrograman Komputer UII. Jumlah data video yang digunakan dalam penelitian ini adalah 30 data, masing-masing 15 data jatuh dan 15 data tidak jatuh. Resolusi video yang dihasilkan adalah 1280 x 720 piksel.

3.1.2 Algoritma Sistem Pendeteksi Jatuh

Data video yang telah diambil sebelumnya akan dianalisa menggunakan algoritma sistem pendeteksi jatuh, yaitu berdasarkan kecepatan pergerakan objek ($C_{motion} > 0.6$) [4], dan perubahan bentuk tubuh objek ($Change\ of\ Sigma_{\rho}\ or\ Sigma_{\theta} < -0.1$). Batasan tersebut dirancang dengan melihat respons parameter dari beberapa data video, kemudian memilih nilai dengan tingkat akurasi tertinggi. Adapun diagram alir algoritma sistem pendeteksi jatuh direpresentasikan pada gambar 3.2 berikut.



Gambar 3.2 Diagram Alir Algoritma Pendeteksi Jatuh.

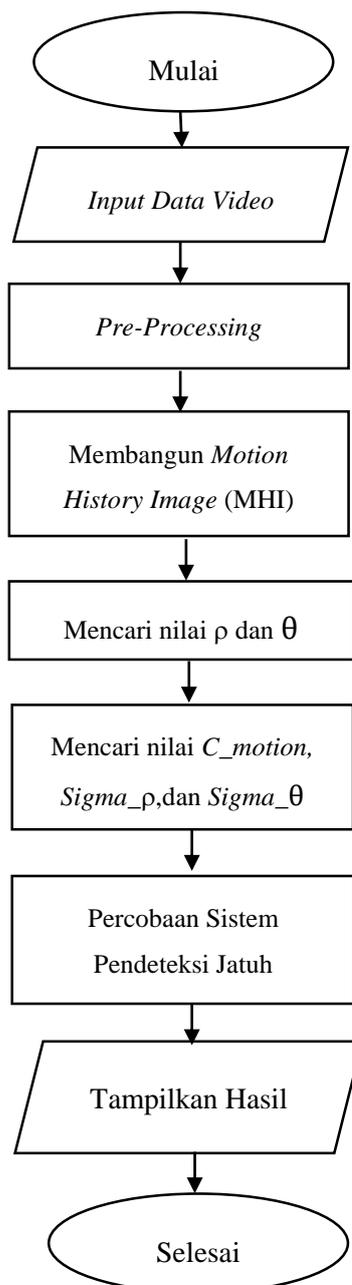
3.1.3 Menampilkan Hasil & Analisa

Setelah semua tahap sudah dilaksanakan maka hasil pendeteksi jatuh akan ditampilkan kemudian dianalisa. Status data input akan dianalisa apakah merupakan data jatuh atau tidak, penentuan tersebut berdasarkan status dari semua kamera CCTV. Ketika “Jatuh terdeteksi pada ≥ 2 kamera” maka data tersebut akan dianggap sebagai data jatuh. Kondisi tersebut digunakan karena

dianggap lebih bagus dari kondisi lain, yaitu ketika menggunakan kondisi “Jatuh terdeteksi ≥ 1 kamera”, status dari kamera lain akan lebih mendominasi. Sedangkan ketika menggunakan kondisi “Jatuh terdeteksi ≥ 3 kamera atau 4 kamera”, akurasi deteksi jatuh menjadi berkurang.

3.2 Sistem Pendeteksi Jatuh

Sistem pendeteksi jatuh dirancang menggunakan *software* MATLAB 2017 dimana setiap tahap memiliki fungsi masing-masing untuk menyelesaikan penelitian ini. Gambar 3.3 merupakan diagram alir sistem.



Gambar 3.3 Diagram Alir Sistem Deteksi Jatuh.

3.2.1 Input Data Video

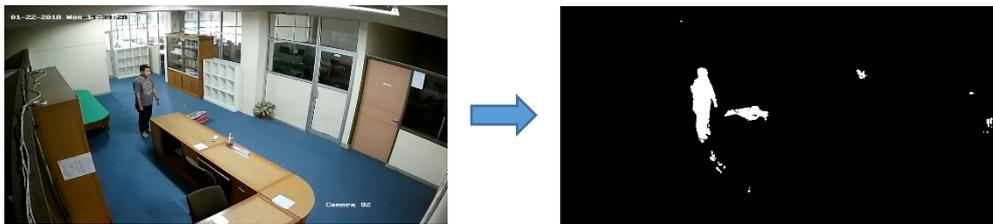
Input dari sistem deteksi jatuh ini adalah data video yang telah direkam pada kamera CCTV sebelumnya, data jatuh maupun tidak jatuh akan dimasukkan secara bergantian pada program matlab untuk membangun parameter yang akan digunakan sebagai acuan pendeteksi jatuh pada sistem.

3.2.2 Pre-Processing

Proses pertama yang dilakukan pada data input video adalah *pre-processing*, proses ini dilakukan sebelum membangun MHI. Adapun proses-proses yang dilakukan terhadap data input video adalah *foreground detection* dan *filtering blob*.

1. *foreground detection*

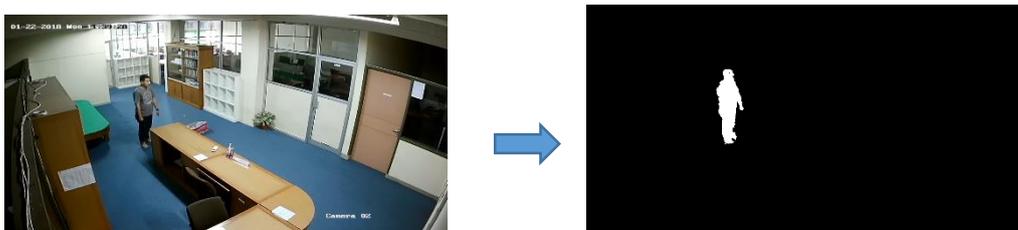
Foreground detection adalah teknik pengolahan citra yang memisahkan objek dengan *background*. Hasil dari deteksi ini adalah citra biner dimana objek yang dideteksi memiliki nilai piksel bernilai 1 (Putih) sedangkan background di sekitar objek memiliki nilai piksel bernilai 0 (hitam) [10]. Gambar 3.4 merupakan gambar hasil ekstraksi objek dengan *foreground detection*.



Gambar 3.4 Hasil *Foreground Detection* pada Sebuah Citra.

2. *filtering blob*

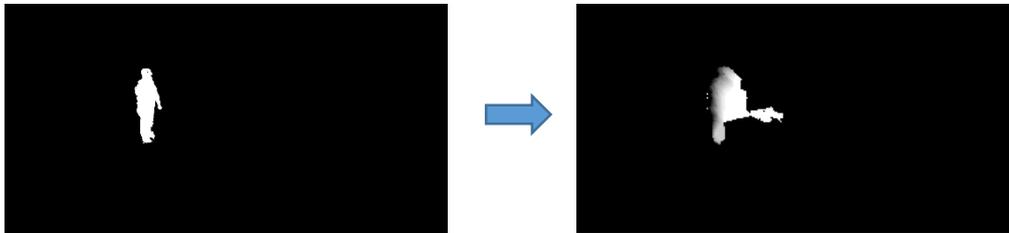
Setelah mendapatkan gambar *foreground* langkah selanjutnya adalah *filtering blob*, tujuan dari proses *filter* ini adalah untuk menghilangkan beberapa *noise* pada citra *foreground* sebelumnya, sehingga beberapa titik-titik putih pada citra menjadi hilang [10]. Gambar 3.5 menunjukkan hasil *filtering blob* pada citra *foreground*.



Gambar 3.5 Hasil *filtering blob* pada citra *foreground*.

3.2.3 Membangun *Motion History Image* (MHI)

Motion history image dibangun menggunakan citra yang telah difilter sebelumnya, kemudian citra tersebut dijadikan data masukan untuk membangun MHI. Data citra akan diolah menggunakan algoritma pada persamaan 2.1 sehingga menghasilkan citra MHI seperti pada gambar 3.6 berikut.



Gambar 3.6 Hasil citra *Motion History Image* (MHI).

3.2.4 Mencari nilai ρ dan ϕ

Setelah mendapatkan MHI, langkah selanjutnya adalah mencari nilai *theta* dan *rho* nilai ini merepresentasikan posisi dari tubuh objek yang dideteksi [4]. Untuk mendapatkan nilai *theta* adalah menggunakan persamaan 2.5, sedangkan untuk mendapatkan nilai *rho* adalah menggunakan persamaan 2.9.

3.2.5 Mencari nilai C_{motion} , Σ_{ρ} , dan Σ_{θ} .

Setelah mendapatkan citra MHI serta parameter ρ dan ϕ , langkah selanjutnya adalah mencari nilai C_{motion} , Σ_{ρ} , dan Σ_{θ} . Nilai ketiga parameter ini merepresentasikan pergerakan serta posisi tubuh objek sehingga ketiga paramet inilah yang akan digunakan sebagai acuan sistem pendeteksi jatuh pada objek. Parameter C_{motion} merepresentasikan kecepatan pergerakan objek yang diberi skala 0-100, dimana 0 berarti diam sedangkan 100 berarti pergerakan penuh [4]. Untuk menghitung nilai C_{motion} adalah menggunakan persamaan berikut.

$$C_{motion} = \frac{\sum_{pixel(x,y) \in blob} H_{\tau}(x,y,t)}{\#pixels \in blob} \quad (3.1)$$

Keterangan :

C_{motion} = Presentase pergerakan objek.

Blob = Piksel yang merepresentasikan objek.

Setelah menghitung C_{motion} maka dihitung Σ_{ρ} dan Σ_{θ} , dimana nilai tersebut didapat dari persamaan berikut.

$$\text{Sigma}_\theta = \sigma_\theta \quad (3.2)$$

$$\text{Sigma}_\rho = \sigma_\rho \quad (3.3)$$

Keterangan :

Sigma_θ = Standar deviasi dari nilai *theta*.

Sigma_ρ = Standar deviasi dari nilai *rho*.

BAB 4

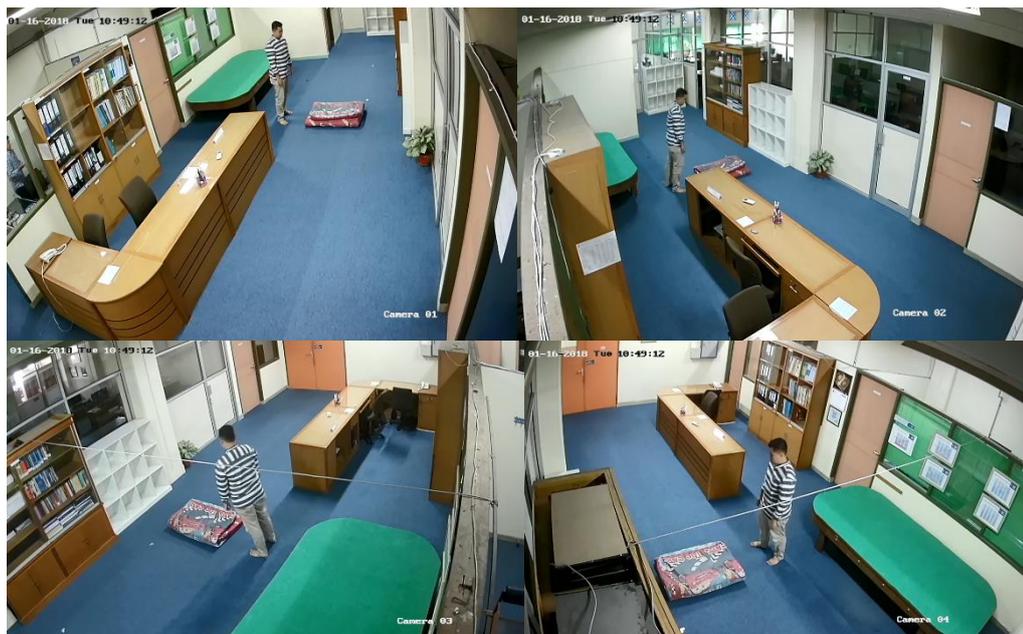
HASIL DAN PEMBAHASAN

4.1 Pengujian Sistem Pendeteksi Jatuh.

Pengujian ini bertujuan untuk mengetahui apakah sistem pendeteksi jatuh dapat berkerja dengan baik. Tahapan dari pengujian ini adalah menguji input kamera CCTV , menghitung parameter objek dengan metode *motion history image* dan *approximated ellipse*, dan melihat hasil kinerja algoritma pendeteksi jatuh.

4.1.1 Pengujian Kamera CCTV

Pada percobaan ini peneliti melakukan perekaman data jatuh objek dengan menggunakan 4 kamera CCTV yang telah dipasang pada setiap sudut ruangan. Gambar 4.1 merupakan tampilan data objek yang telah direkam menggunakan semua CCTV.



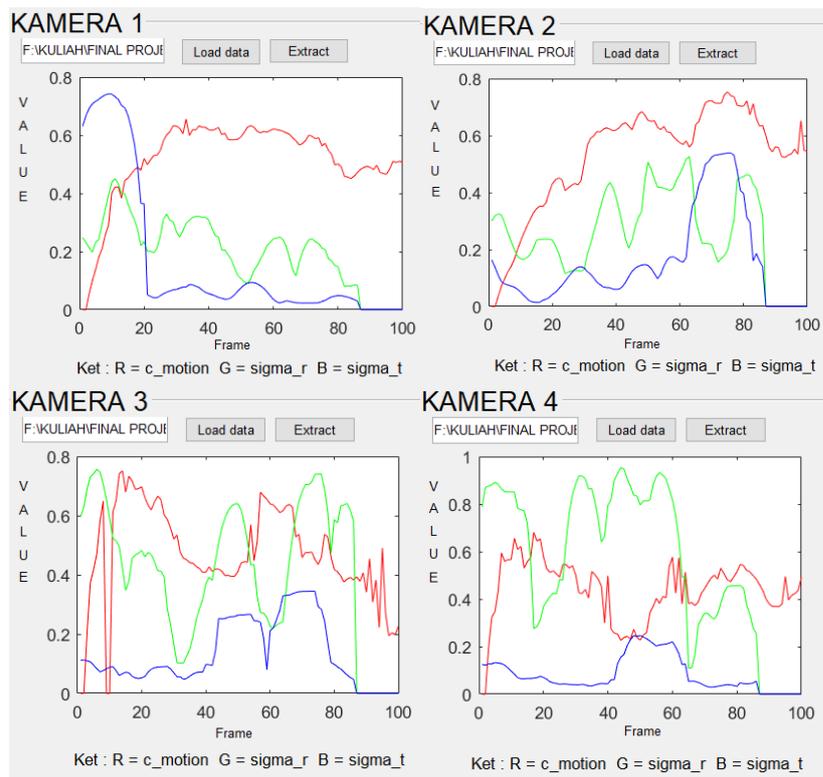
Gambar 4.1 Tampilan Objek dari Kamera CCTV.

Gambar 4.1 menunjukkan bahwa semua kamera CCTV dapat merekam data objek sehingga dengan data yang direkam tersebut maka dapat dilanjutkan pada proses selanjutnya yaitu membangun parameter C_{motion} , $Sigma_{theta}$ dan $Sigma_{rho}$.

4.1.2 Pengujian Perhitungan Parameter Pergerakan Objek

Pada percobaan ini peneliti membangun parameter objek menggunakan metode *motion history image* dan *approximated ellipse*, parameter ini akan digunakan sebagai data acuan

percobaan algoritma pendeteksi jatuh yang akan digunakan pada langkah selanjutnya. Gambar 4.2 menunjukkan hasil parameter yang telah dibangun.



Gambar 4.2 Hasil Parameter Sistem Pendeteksi Jatuh.

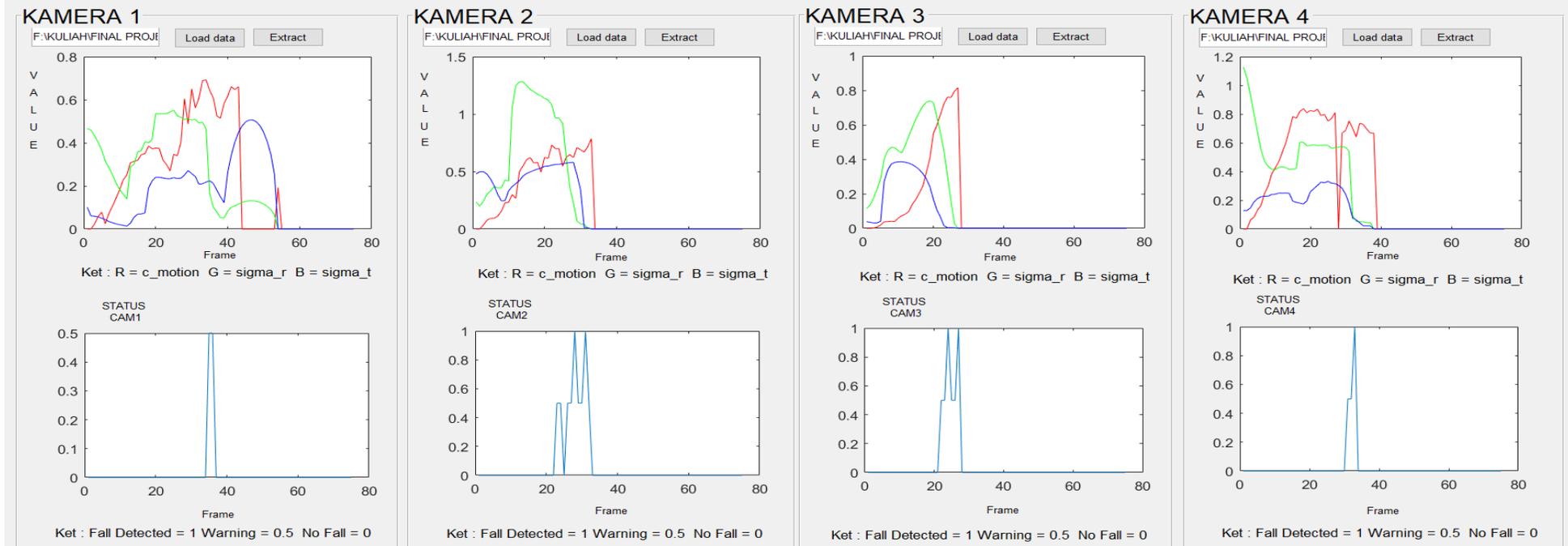
Gambar 4.2 menunjukkan bahwa parameter dapat dibangun dengan menggunakan metode *motion history image* dan *approximated ellipse*. Hasil dari semua parameter ini akan dijadikan acuan algoritma untuk mendeteksi apakah terdapat gerakan jatuh pada objek atau tidak.

4.1.3 Pengujian Algoritma Peneteksi Jatuh

Pengujian ini bertujuan untuk mengetahui apakah algoritma pendeteksi jatuh objek dapat bekerja. Algoritma pendeteksi jatuh menggunakan data input parameter yang dihasilkan oleh proses sebelumnya. Gambar 4.3 menunjukkan hasil dari sistem pendeteksi jatuh menggunakan metode MHI dan *approximated elli*

PENDETEKSI JATUH MENGGUNAKAN PEMROSESAN CITRA

MOHD BRADO FRASETYO (14524030)



Gambar 4.3 Tampilan Hasil Sistem Pendeteksi Jatuh.

Pada gambar 4.3 menunjukkan bahwa sistem dapat menghasilkan hasil deteksi jatuh yang baik, pada percobaan ini menggunakan satu data jatuh kemudian menunjukkan bahwa 3 kamera dapat mendeteksi jatuh pada objek yang berarti data yang dimasukkan merupakan data pergerakan jatuh.

4.2 Pengujian Algoritma Terhadap Kondisi Berbeda.

4.2.1 Pengujian Algoritma Pendeteksi Jatuh Terhadap Posisi Jatuh yang Berbeda.

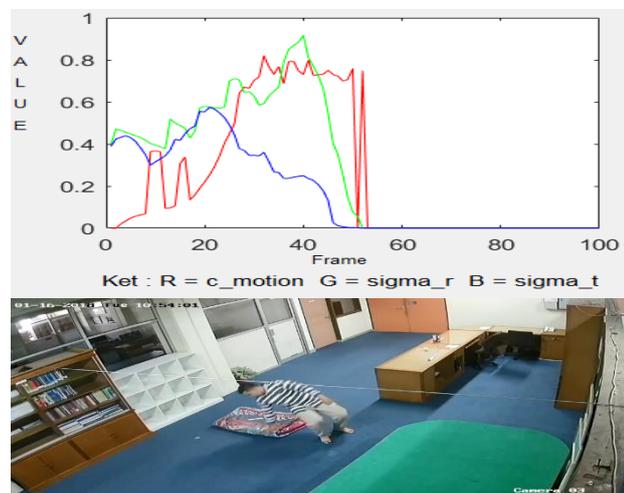
Pengujian ini bertujuan untuk mengetahui apakah posisi jatuh objek berpengaruh terhadap algoritma deteksi jatuh. Perbedaan posisi objek ini berhubungan dengan jarak antara objek terhadap masing-masing kamera serta keberadaan benda lain yang menghalangi objek, sehingga percobaan dilakukan dengan mengambil data jatuh pada posisi berbeda. Hasil percobaan dapat dilihat pada tabel 4.1.

Tabel 4.1 Hasil pengujian pendeteksi jatuh terhadap posisi jatuh objek yang berbeda.

NO	Posisi	Kamera 1	Kamera 2	Kamera 3	Kamera 4
1	Tengah Ruangan	✓	✓	✓	✓
2	Berdekatan dengan Kamera 1	✗	✓	✓	✓
3	Berdekatan dengan Kamera 2	✓	✓	✗	✗
4	Berdekatan dengan Kamera 3	✓	✗	✗	✓
5	Berdekatan dengan Kamera 4	✗	✗	✓	✗

Keterangan : ✓ = Jatuh Terdeteksi ; ✗ = Jatuh Tidak Terdeteksi

Pada tabel 4.1 menunjukkan bahwa posisi objek yang berada di tengah ruangan merupakan posisi yang paling ideal untuk mendeteksi jatuh, hal ini dikarenakan semua pergerakan dari objek dapat terlihat jelas oleh semua kamera, sehingga data parameter yang didapat menjadi lebih akurat. Hasil dapat dilihat pada Gambar 4.4 yang menunjukkan parameter jatuh yang terdeteksi dari salah satu data pada kamera 3.

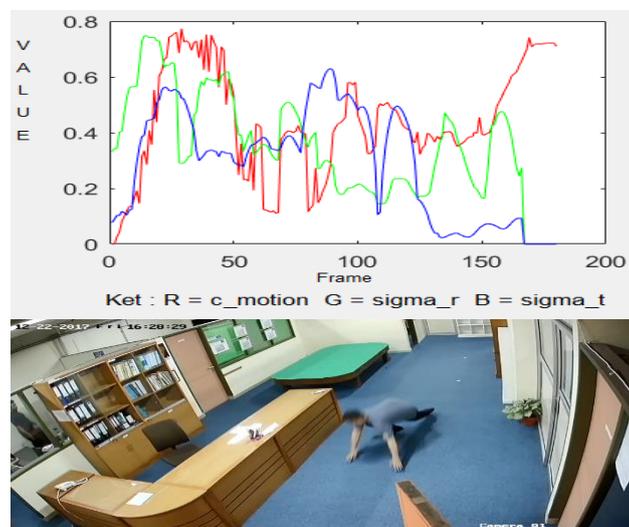


Gambar 4.4 Perubahan parameter ketika objek berada di posisi tengah ruangan.

Selain posisi tengah ruangan yang dapat dideteksi dengan baik oleh sistem, terdapat beberapa posisi yang sulit terdeteksi yaitu jarak antara kamera dengan objek, dan keberadaan objek lain yang menghalangi pergerakan objek. Berikut analisa beberapa posisi objek yang berpengaruh terhadap performa algoritma pendeteksi jatuh.

1. Posisi Objek Berdekatan Dengan Kamera

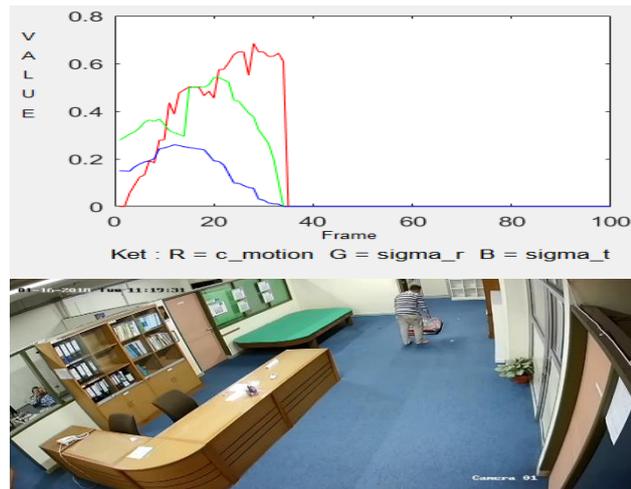
Hasil pengujian ini menunjukkan bahwa ketika objek berada sangat dekat dengan kamera, akurasi algoritma pendeteksi jatuh menjadi berkurang. Hal ini karena semua gerakan termasuk bayangan dari objek akan terdeteksi dan menghasilkan perubahan parameter yang tidak stabil dan algoritma sistem tidak dapat mendeteksi jatuh dengan baik. Hasil dapat dilihat pada Gambar 4.5 yang menunjukkan perubahan parameter yang tidak stabil.



Gambar 4.5 Perubahan parameter ketika objek dekat dengan kamera.

2. Posisi Objek Jauh dari Kamera

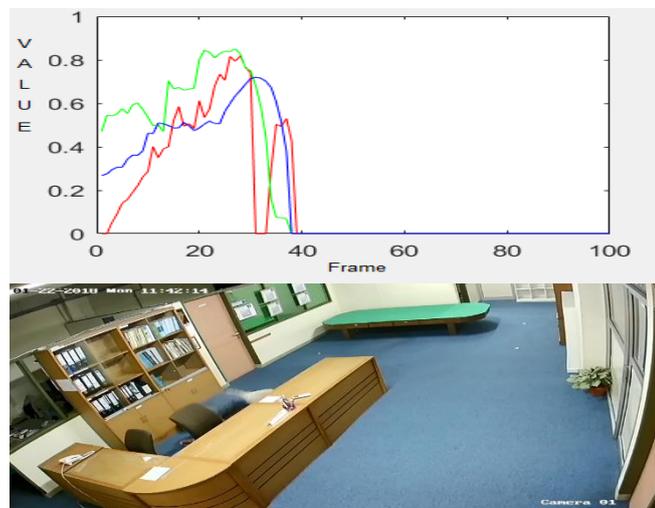
Hasil pengujian ini menunjukkan bahwa ketika objek berada terlalu jauh dari kamera, akurasi algoritma pendeteksi jatuh menjadi berkurang. Hal ini dikarenakan objek berada terlalu jauh yang menyebabkan data perubahan piksel hanya pada area kecil saja. Ketika jatuh terjadi, data *C_motion* mengalami perubahan yang sangat cepat sehingga algoritma hanya menghasilkan peringatan jatuh, dan tidak mendeteksi jatuh secara utuh. Hasil dapat dilihat pada Gambar 4.6 yang menunjukkan perubahan parameter *C_motion* yang sangat cepat.



Gambar 4.6 Perubahan parameter ketika jarak terlalu jauh.

3. Posisi Objek Terhalang Objek Lain.

Hasil pengujian ini menunjukkan bahwa ketika objek berada dibelakang objek lain, akurasi algoritma pendeteksi jatuh menjadi berkurang. Hal ini dikarenakan kamera tidak dapat mendeteksi pergerakan objek secara keseluruhan, sehingga parameter C_motion menjadi terpotong. Hasil dapat dilihat pada Gambar 4.7 yang menunjukkan parameter yang terpotong.



Gambar 4.7 Perubahan parameter ketika objek terhalang benda lain.

4.2.2 Pengujian Algoritma Pendeteksi Jatuh Terhadap Gerakan Jatuh yang Berbeda

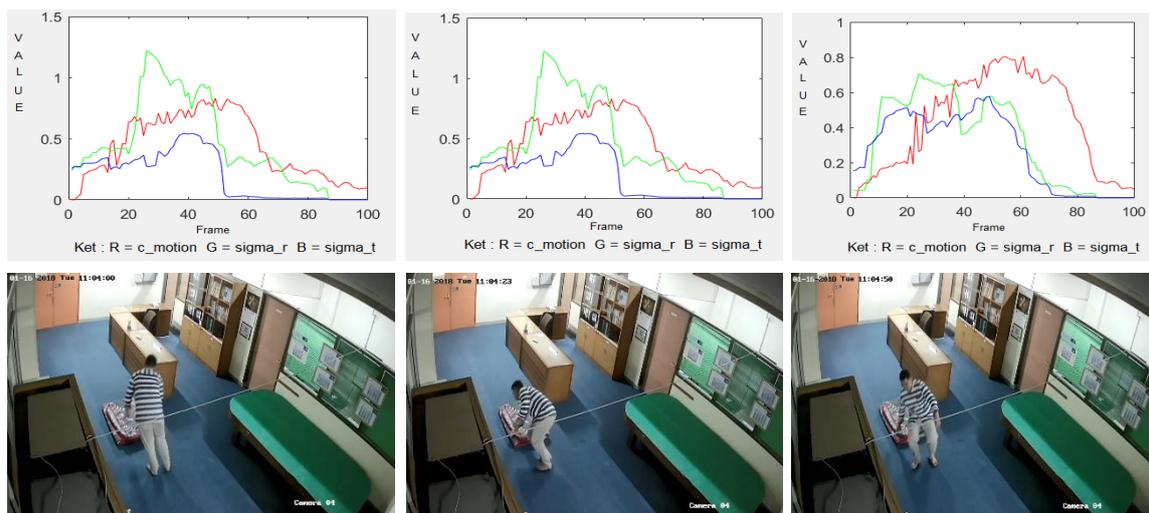
Pengujian ini bertujuan untuk mengetahui apakah perbedaan pergerakan jatuh berpengaruh terhadap performa algoritma deteksi jatuh, hal ini dikarenakan pergerakan jatuh dapat berbeda-beda seperti depan, samping dan belakang. Hasil percobaan dapat dilihat pada tabel 4.2.

Tabel 4.2 Hasil pengujian pendeteksi jatuh terhadap pergerakan jatuh yang berbeda.

NO	Gerak Jatuh	Kamera 1	Kamera 2	Kamera 3	Kamera 4
1	Gerak Jatuh ke Depan (1)	✘	✘	✓	✓
2	Gerak Jatuh ke Depan (2)	✓	✓	✓	✘
3	Gerak Jatuh ke Depan (3)	✘	✓	✘	✓
4	Gerak Jatuh ke Samping (1)	✓	✓	✘	✓
5	Gerak Jatuh ke Samping (2)	✘	✓	✓	✓
6	Gerak Jatuh ke Samping (3)	✘	✘	✓	✓
7	Gerak Jatuh ke Belakang (1)	✓	✓	✓	✘
8	Gerak Jatuh ke Belakang (2)	✘	✓	✓	✘
9	Gerak Jatuh ke Belakang (3)	✘	✓	✓	✓

Keterangan : ✓ = Jatuh Terdeteksi ; ✘ = Jatuh Tidak Terdekteksi

Dari table 4.2 menunjukkan bahwa semua pergerakan jatuh dapat terdeteksi menggunakan algoritma pendeteksi jatuh. Parameter yang dihasilkan cenderung sama walaupun pergerakan jatuh yang berbeda, karena pergerakan jatuh rata-rata memiliki kecenderungan pergerakan yang mirip yaitu berawal dari posisi berdiri, bergerak secara cepat, tertidur kemudian diam. Perbandingan parameter dapat dilihat pada Gambar 4.8 berikut.



Gambar 4.8 Perubahan parameter ketika jatuh arah depan, samping, dan belakang.

4.2.3 Pengujian Algoritma Pendeteksi Jatuh Terhadap Intensitas Cahaya Berbeda.

Pengujian ini bertujuan untuk mengetahui bagaimana pengaruh intensitas cahaya terhadap performa algoritma deteksi jatuh. Pengambilan data untuk percobaan ini dilakukan dengan menggunakan intensitas cahaya terang, redup dan cahaya gelap. Hasil percobaan dapat dilihat pada table 4.3.

Tabel 4.3 Hasil pengujian pendeteksi jatuh terhadap intensitas cahaya yang berbeda.

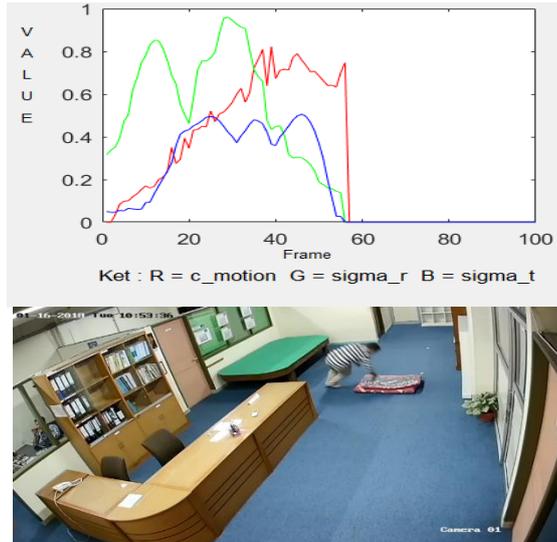
NO	Intensitas Cahaya	Kamera 1	Kamera 2	Kamera 3	Kamera 4
1	Cahaya Terang (1)	✘	✘	✓	✓
2	Cahaya Terang (2)	✓	✓	✓	✘
3	Cahaya Terang (3)	✓	✓	✘	✓
4	Cahaya Redup (1)	✘	✓	✘	✓
5	Cahaya Redup (2)	✘	✘	✓	✓
6	Cahaya Redup (3)	✘	✓	✓	✓
7	Cahaya Gelap (1)	✘	✘	✘	✘
8	Cahaya Gelap (1)	✘	✘	✘	✘

Keterangan : ✓ = Jatuh Terdeteksi ; ✘ = Jatuh Tidak Terdekteksi

Pada tabel 4.3 menunjukan bahwa intensitas cahaya sangat berpengaruh terhadap performa algoritma pendeteksi jatuh, berikut penjelasan tentang pengaruh intensitas cahaya terhadap sistem.

1. Cahaya Terang

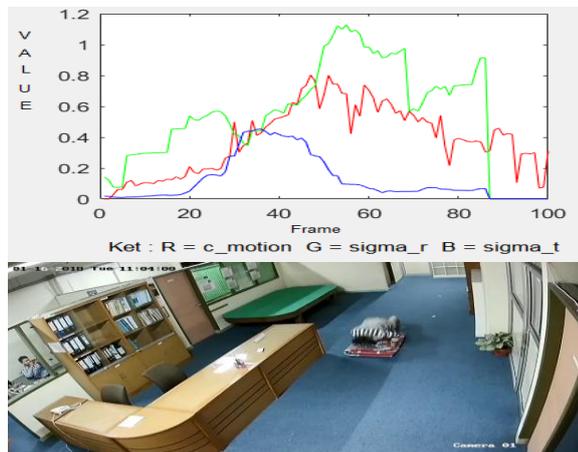
Hasil pengujian pada percobaan ini menunjukan bahwa ketika cahaya terang algoritma dapat mendeteksi jatuh dengan baik, hal ini dikarenakan kamera dapat menangkap perubahan parameter dengan lebih jelas dan hanya mendapat sedikit gangguan dari bayangan objek, sehingga parameter yang dihasilkan menjadi lebih akurat. Gambar 4.9 menunjukan hasil percobaan dari cahaya terang.



Gambar 4.9 Perubahan parameter ketika intensitas cahaya terang.

2. Cahaya Redup

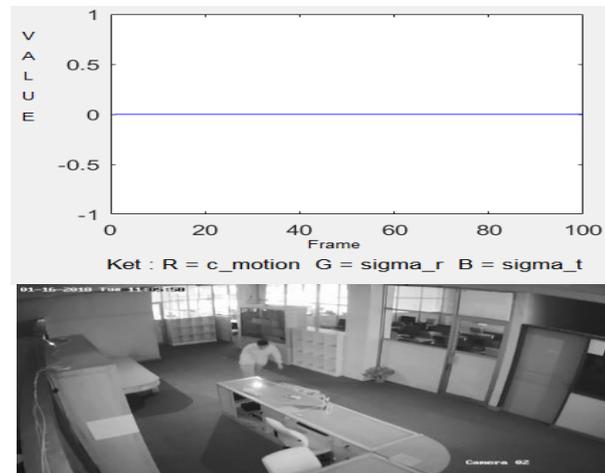
Hasil pengujian pada percobaan ini menunjukkan bahwa ketika cahaya redup maka algoritma masih dapat mendeteksi jatuh dari objek, tetapi terdapat salah satu kamera yang tidak dapat mendeteksi jatuh yaitu kamera 1. Hal ini dikarenakan kamera 1 mendapat gangguan bayangan yang lebih banyak dari kamera yang lain sehingga nilai parameter menjadi terganggu, tetapi untuk kamera yang lain algoritma masih dapat mendeteksi jatuh objek dengan baik. Gambar 4.10 menunjukkan hasil percobaan cahaya redup dengan gangguan bayangan.



Gambar 4.10 Perubahan parameter ketika intensitas cahaya redup.

3. Cahaya Gelap

Hasil pengujian pada percobaan ini menunjukkan bahwa ketika cahaya dalam kondisi gelap maka jatuh tidak dapat terdeteksi. Hal ini dikarenakan data piksel tidak tertangkap oleh kamera sehingga parameter C_motion tidak muncul. Gambar 4.11 menunjukkan hasil percobaan cahaya gelap.



Gambar 4.11 Perubahan parameter ketika intensitas cahaya gelap.

4.2.4 Pengujian Algoritma Pendeteksi Jatuh Terhadap Warna Objek Berbeda

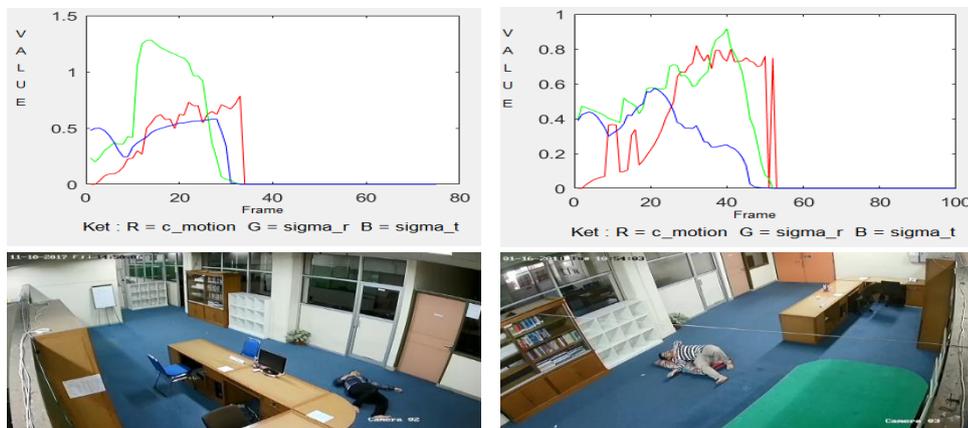
Pengujian ini bertujuan untuk mengetahui bagaimana pengaruh perbedaan warna objek dengan *background* terhadap performa algoritma deteksi jatuh. Pengambilan data untuk percobaan ini dilakukan dengan menggunakan warna yang sama dan berbeda dengan *background*. Hasil percobaan dapat dilihat pada tabel 4.4.

Tabel 4.4 Hasil pengujian pendeteksi jatuh dengan warna objek yang berbeda.

NO	Warna Objek	Kamera 1	Kamera 2	Kamera 3	Kamera 4
1	Warna Sama (1)	✘	✓	✓	✓
2	Warna Sama (2)	✓	✓	✓	✘
3	Warna Sama (3)	✘	✓	✓	✓
4	Warna Berbeda (1)	✘	✘	✓	✓
5	Warna Berbeda (2)	✓	✓	✘	✓
6	Warna Berbeda (3)	✘	✓	✓	✘

Keterangan : ✓ = Jatuh Terdeteksi ; ✘ = Jatuh Tidak Terdekteksi

Hasil pengujian pada percobaan ini menunjukkan bahwa semua kamera dapat mendeteksi objek yang menggunakan pakaian yang mempunyai warna yang identik sama dengan warna *background*. Hal ini dikarenakan parameter yang dihasilkan tidak berdasarkan warna melainkan berdasarkan pergerakan dari objek tersebut sehingga algoritma bisa mendeteksi objek yang jatuh dengan baik. Gambar 4.12 menunjukkan hasil parameter dengan warna sama dan berbeda.



Gambar 4.12 Perubahan parameter ketika warna objek berbeda.

4.2.5 Pengujian Algoritma Pendeteksi Jatuh Terhadap Objek Berbeda

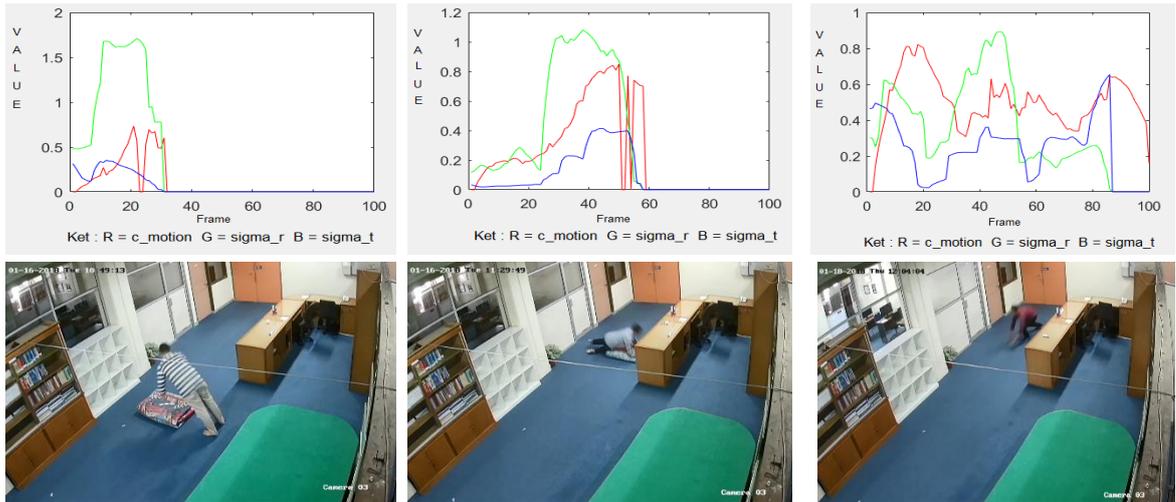
Pengujian ini bertujuan untuk mengetahui bagaimana pengaruh perbedaan objek yang ditangkap kamera terhadap performa algoritma deteksi jatuh. Pengambilan data untuk percobaan ini dilakukan dengan mengambil data 3 objek berbeda. Hasil percobaan dapat dilihat pada tabel 4.5.

Tabel 4.5 Hasil pengujian pendeteksi jatuh terhadap objek yang berbeda.

NO	Objek	Kamera 1	Kamera 2	Kamera 3	Kamera 4
1	Objek 1	✓	✓	✓	✓
2	Objek 2	✓	✓	✓	✓
3	Objek 3	✗	✓	✓	✗

Keterangan : ✓ = Jatuh Terdeteksi ; ✗ = Jatuh Tidak Terdekteksi

Hasil pengujian ini menunjukkan bahwa kamera dapat mendeteksi objek yang berbeda, pengujian dilakukan dengan mengambil beberapa data. Untuk objek 3 percobaan hanya dilakukan 1 kali tetapi algoritma masih dapat mendeteksi jatuh pada objek. Gambar 4.13 menunjukkan hasil percobaan dengan objek berbeda.



Gambar 4.13 Perubahan parameter ketika objek berbeda.

4.2.6 Pengujian Algoritma Pendeteksi Jatuh Terhadap Kecepatan Jatuh Berbeda.

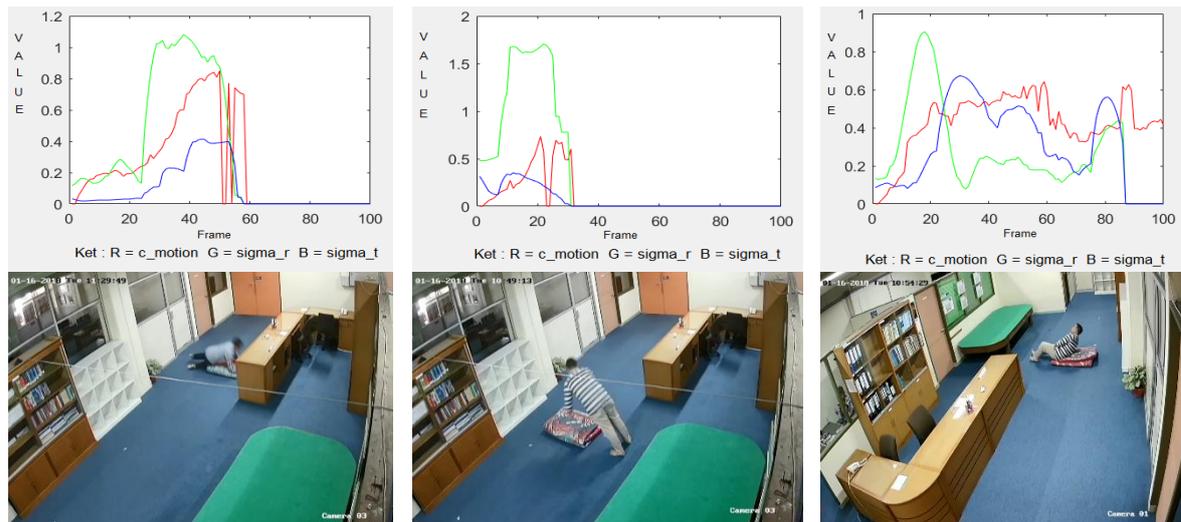
Pengujian ini bertujuan untuk mengetahui bagaimana pengaruh perbedaan kecepatan jatuh objek terhadap performa algoritma deteksi jatuh. Pengambilan data untuk percobaan ini dilakukan dengan variasi kecepatan yaitu cepat, sedang, dan lambat. Hasil percobaan dapat dilihat pada table 4.6.

Tabel 4.6 Hasil pengujian pendeteksi jatuh terhadap kecepatan jatuh yang berbeda.

NO	Kecepatan	Kamera 1	Kamera 2	Kamera 3	Kamera 4
1	Cepat	✓	✓	✓	✓
2	Sedang	✓	✓	✓	✓
3	Lambat	✗	✗	✗	✗

Keterangan : ✓ = Jatuh Terdeteksi ; ✗ = Jatuh Tidak Terdekteksi

Hasil pengujian pada percobaan ini menunjukkan bahwa kecepatan jatuh sangat berpengaruh terhadap performa pendeteksi jatuh. Hal ini dikarenakan kecepatan berpengaruh terhadap perubahan parameter C_motion . Apabila kecepatan cepat dan sedang maka nilai C_motion akan lebih dari 0.6 sehingga jatuh akan terdeteksi, sedangkan apabila kecepatan terlalu lambat maka tidak dapat menghasilkan deteksi jatuh yang sesuai karena nilai dari C_motion akan kurang dari 0.6. Gambar 4.14 merupakan perbandingan perubahan parameter antara kecepatan jatuh cepat, sedang dan lambat.



Gambar 4.14 Perubahan parameter ketika kecepatan jatuh berbeda.

4.3 Pengujian Akurasi Sistem Pendeteksi Jatuh.

Pengujian ini bertujuan untuk menghitung akurasi sistem pendeteksi jatuh, pengujian dilakukan dengan cara mengambil banyak data yaitu dengan kondisi berbeda. Perbedaan data yang digunakan adalah perbedaan dalam posisi, pergerakan jatuh, intensitas cahaya, warna objek dengan *background*, kecepatan, objek berbeda, dan data bukan jatuh. Tabel 4.7 merupakan hasil pengujian akurasi jatuh dengan menggunakan 30 data.

Tabel 4.7 Hasil pengujian akurasi sistem pendeteksi jatuh.

NO	Data	Kamera 1	Kamera 2	Kamera 3	Kamera 4
1	Jatuh (1)	✘	✘	✓	✓
2	Jatuh (2)	✓	✓	✘	✓
3	Jatuh (3)	✓	✓	✓	✘
4	Jatuh (4)	✘	✓	✓	✓
5	Jatuh (5)	✘	✓	✘	✓
6	Jatuh (6)	✘	✘	✓	✓
7	Jatuh (7)	✘	✓	✓	✓
8	Jatuh (8)	✓	✓	✓	✘
9	Jatuh (9)	✘	✓	✓	✘
10	Jatuh (10)	✘	✘	✓	✓

NO	Data	Kamera 1	Kamera 2	Kamera 3	Kamera 4
11	Jatuh (11)	✘	✓	✓	✘
12	Jatuh (12)	✘	✓	✓	✘
13	Jatuh (13)	✘	✓	✘	✘
14	Jatuh (14)	✓	✘	✘	✓
15	Jatuh (15)	✘	✓	✓	✓
16	Bukan Jatuh (1)	✘	✘	✘	✘
17	Bukan Jatuh (2)	✘	✓	✓	✓
18	Bukan Jatuh (3)	✘	✓	✘	✘
19	Bukan Jatuh (4)	✓	✓	✓	✓
20	Bukan Jatuh (5)	✘	✘	✘	✘
21	Bukan Jatuh (6)	✓	✘	✘	✘
22	Bukan Jatuh (7)	✘	✘	✓	✓
23	Bukan Jatuh (8)	✘	✘	✘	✘
24	Bukan Jatuh (9)	✘	✘	✘	✘
25	Bukan Jatuh (10)	✘	✓	✘	✓
26	Bukan Jatuh (11)	✘	✘	✘	✘
27	Bukan Jatuh (12)	✘	✘	✘	✘
28	Bukan Jatuh (13)	✘	✘	✘	✘
29	Bukan Jatuh (14)	✘	✘	✘	✓
30	Bukan Jatuh (15)	✘	✘	✘	✘

Keterangan : ✓ = Jatuh Terdeteksi ; ✘ = Jatuh Tidak Terdekteksi

Hasil dari pengujian dengan 30 data menunjukkan bahwa akurasi sistem adalah sebesar 83.33%. Terdapat kesalahan pada data tidak jatuh yaitu ketika objek berjalan cepat kemudian berhenti, dan objek duduk dengan kecepatan tinggi, pergerakan tersebut menyebabkan parameter mengalami perubahan mirip dengan jatuh.

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan percobaan yang telah dilakukan, maka didapatkan kesimpulan berikut :

1. Sistem pendeteksi jatuh dipengaruhi oleh beberapa kondisi yang menyebabkan performa algoritma menjadi terganggu, yaitu :
 - a. Jarak antara kamera dengan objek terlalu jauh ataupun terlalu dekat.
 - b. Keberadaan objek lain yang menghalangi objek yang akan dideteksi.
 - c. Kecepatan jatuh objek.
 - d. Intensitas cahaya ruangan.
2. Sistem pendeteksi jatuh memiliki performa yang baik ketika kondisi berikut :
 - a. Perbedaan pergerakan jatuh objek.
 - b. Perbedaan warna antara objek dengan *background* yang kontras.
3. Sistem pendeteksi jatuh pada penelitian ini memiliki akurasi sebesar 83.33%, dengan menggunakan percobaan data jatuh dan tidak jatuh yang bervariasi.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan, penulis memiliki beberapa saran untuk penelitian kedepannya agar performa dari sistem menjadi lebih baik lagi, yaitu :

1. Penambahan filter lain dalam membangun parameter *motion history image* (C_{motion}) agar bayangan dari objek tidak terdeteksi dengan mudah, sehingga parameter yang dihasilkan memang berdasarkan dari objek yang akan dideteksi tanpa adanya gangguan dari piksel yang lain.
2. Algoritma yang digunakan masih memiliki akurasi yang rendah terhadap pergerakan seperti duduk dengan kecepatan cepat, serta jalan cepat yang kemudian tiba-tiba berhenti. Sebaiknya penelitian selanjutnya dapat memperbaiki algoritma yang memiliki akurasi yang lebih baik dari penelitian ini.
3. Melakukan kompresi citra, karena data yang dihasilkan untuk mendapatkan parameter C_{Motion} , $Sigma_{theta}$, dan $Sigma_{rho}$ membutuhkan kapasitas memori yang besar.

DAFTAR PUSTAKA

- [1] Badan Pusat Statistik. "Statistik Penduduk Lanjut Usia." *Badan Pusat Statistik, Jakarta – Indonesia*, 2015.
- [2] Kementerian Kesehatan, R. I. "Situasi Lanjut Usia (Lansia) di Indonesia: 29 Mei–Hari Lanjut Usia Nasional." *InfoDATIN Pusat Data dan Informasi Kementerian Kesehatan RI*, 2016.
- [3] Kwolek, Bogdan, and Michal Kepski. "Improving fall detection by the use of depth sensor and accelerometer." *Neurocomputing*. Vol. 168, 2015.
- [4] Rougier, Caroline, et al. "Fall detection from human shape and motion history using video surveillance." *Advanced Information Networking and Applications Workshops, 2007, AINAW'07. 21st International Conference on*. Vol. 2. IEEE, 2007.
- [5] Statistik, Badan Pusat. "Kewarganegaraan, suku bangsa, agama, dan bahasa sehari-hari penduduk Indonesia: Hasil sensus penduduk 2010." *Badan Pusat Statistik, Jakarta – Indonesia*, 2011.
- [6] Worrakulpanit, Nuttapong, and Pranchalee Samanpiboon. "Human fall detection using standard deviation of C-motion method." *Journal of Automation and Control Engineering*. Vol.2 No.4, 2014.
- [7] Albawendi, Suad, et al. "Video based fall detection with enhanced motion history images." *Proceedings of the 9th ACM International Conference on Pervasive Technologies Related to Assistive Environments*. ACM, 2016.
- [8] T.Sutoyo, Edy Mulyanto dkk. "Teori Pengolahan Citra Digital" *Andi & Universitas Dian Nuswantoro- Semarang*, 2009.
- [9] Bobick, Aaron F., and James W. Davis. "The recognition of human movement using temporal templates." *IEEE Transactions on pattern analysis and machine intelligence*. Vol.23 No.3, 2001.
- [10] Kim, Kyungnam, et al. "Real-time foreground–background segmentation using codebook model." *Real-time imaging*. Vol.11 No.3, 2005.

LAMPIRAN

1. List Program Membangun Parameter.

```
clc
clear all
disp('Loading Video..');

a=VideoReader('Test Videos\tidak_jatuh10_cam4.avi');

jumlahFrame = 99;
pixelX = 1280;
pixelY = 720;

%buat Cell
frames = cell(1,jumlahFrame);

for img = 1:jumlahFrame
    b = read(a,img);
    frames(1,img) = mat2cell(b,pixelY,pixelX,3);
end

disp('Extracting Foreground..');
fg = extractForeground(frames,30,4,0);
disp('Eliminating Noise from Foreground..');
fg_smooth=detectBlob(fg);
disp('Evaluating Motion History Image of the video..');
motion_history = MHI(fg_smooth);
disp('Determining Ellipse Statistical Properties..');
c = centroid(fg_smooth);
[thetas rhos] = OrientEccent(fg_smooth, c);
disp('Analyzing Statistics for Video..');
[sigma_t sigma_r c_motion] = statistics_2(thetas, rhos,
motion_history , pixelX, pixelY);
subplot(2,2,1), plot(sigma_t),title('Theta Std Dev Values'),
subplot(2,2,2), plot(sigma_r),title('Rho Std Dev Values'),
subplot(2,2,3), plot(c_motion),title('C Motion Values');

viewAllStatistic(1,99, c_motion, sigma_t,sigma_r,
motion_history);

disp('Done!!!');
```

2. List Program Algoritma Deteksi Jatuh Pada GUI

```
%% Algorithm Motion
for frameIndex = 1:lengthFrame
    MotionNote(frameIndex).frame = motion_history(frameIndex);
    MotionNote(frameIndex).c_motion = c_motion(frameIndex);
    MotionNote(frameIndex).sigma_t = sigma_t(frameIndex);
```

```

MotionNote(frameIndex).sigma_r = sigma_r(frameIndex);

if(frameIndex == 1 || frameIndex == 2 )
    MotionNote(frameIndex).changeofsigma_r = 0;
else
    MotionNote(frameIndex).changeofsigma_r =
sigma_r(frameIndex)-sigma_r(frameIndex-2);
end

if(frameIndex == 1 || frameIndex == 2)
    MotionNote(frameIndex).changeofsigma_t = 0;
else
    MotionNote(frameIndex).changeofsigma_t =
sigma_t(frameIndex)-sigma_t(frameIndex-2);
end

% Fall Detect Algorithm : Method Treshold
if(c_motion(frameIndex) > 0.6 &&(
MotionNote(frameIndex).changeofsigma_r < (-0.1) ||
MotionNote(frameIndex).changeofsigma_t < (-0.1)))
    MotionNote(frameIndex).note = 'Warning';
    framemin1 = frameIndex-1;
    framemin2 = frameIndex-2;
    status(frameIndex) = 0.5;
    if(framemin1 > 0 && framemin2 > 0)
        if(strcmp(MotionNote(framemin1).note, 'Warning') &&
strcmp(MotionNote(framemin2).note, 'Warning') )
            MotionNote(frameIndex).note = 'FALL DETECTED';
            status(frameIndex) = 1;
        end
    end
end

else
    MotionNote(frameIndex).note = 'Aman';
    status(frameIndex) = 0;
end

if strcmp(MotionNote(frameIndex).note, 'FALL DETECTED')
    fprintf('WARNING!! FALL DETECTED!! \n');
end

end
end

```